Unpackers in a World of Signature-less Malware Detection

Frederic Vila
Malware Researcher
September 2013

ThreatTrack Security

INTRODUCTION

Approaches to Detection

Signature based

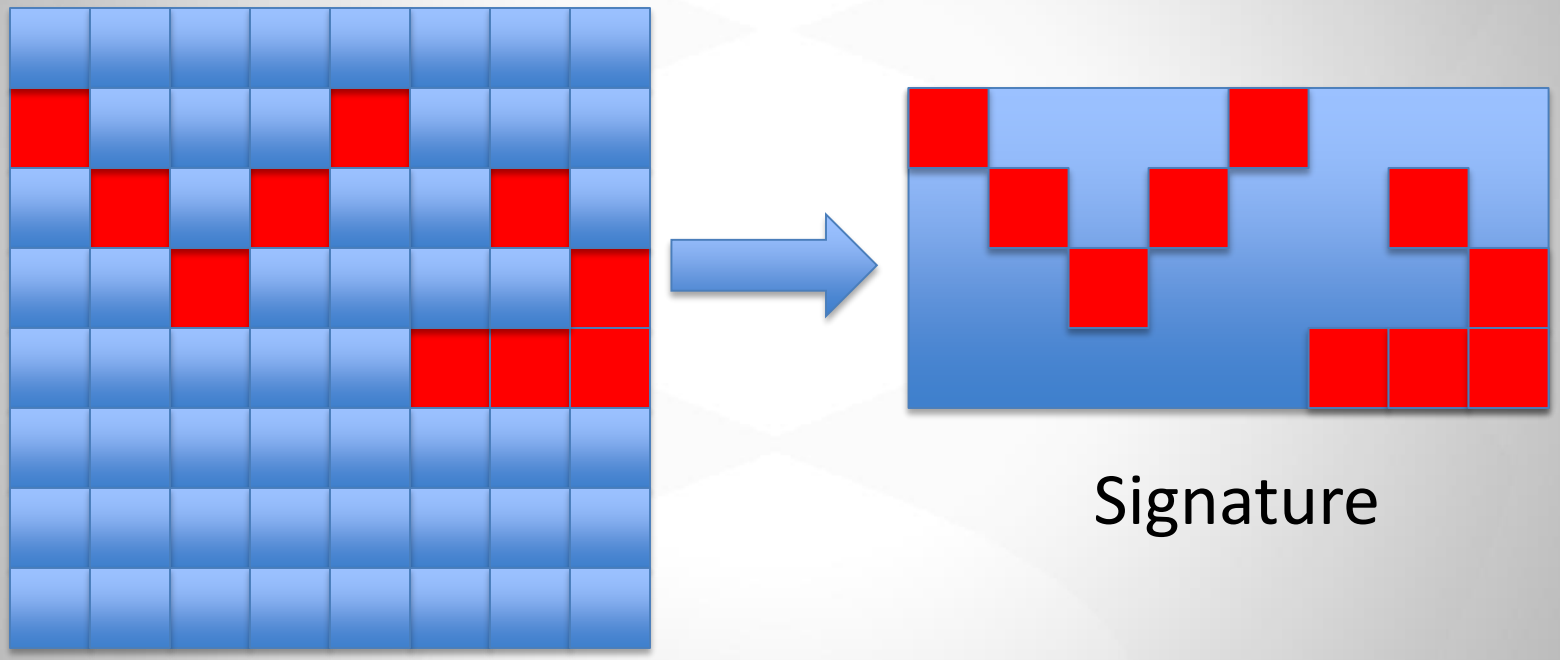
Anomaly based

Hybrid

Behaviour based

Specification based

Signature Based Detection

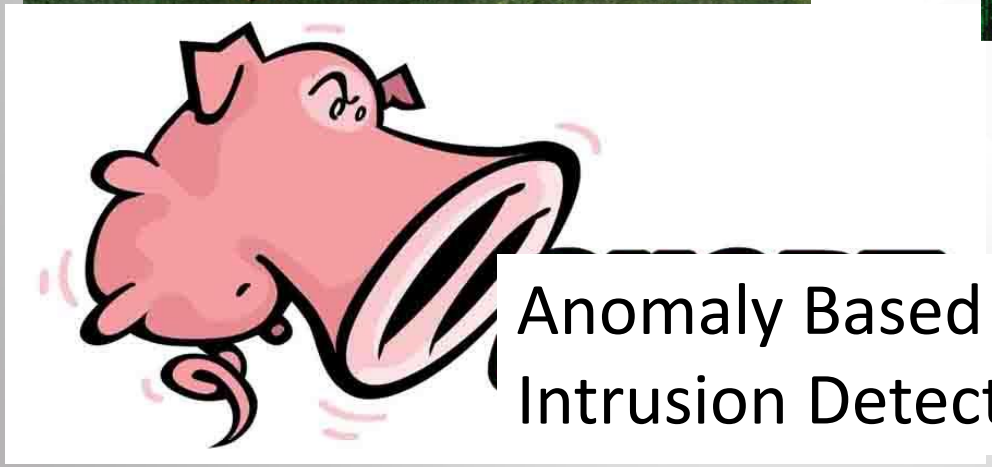


Application

Signature-less Based Detection

Sandbox Software

Virtualization Engine



Anomaly Based
Intrusion Detection System

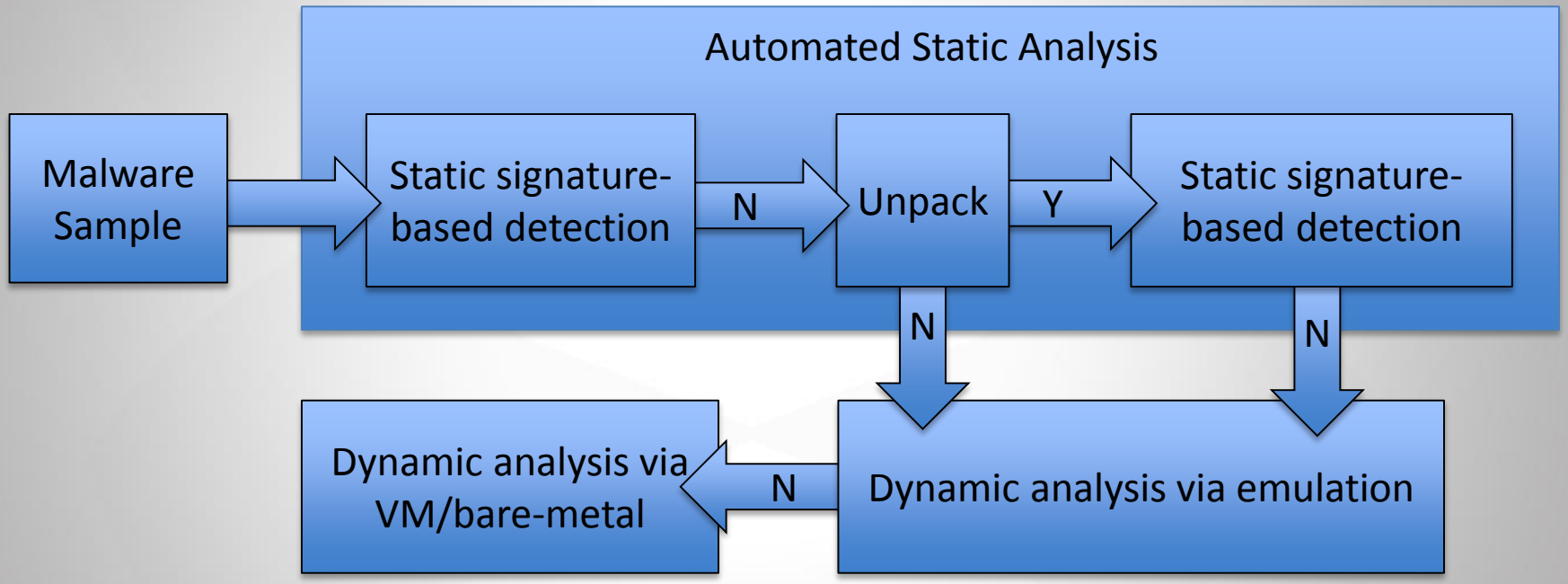
Next Generation Anti-Virus (NGAV)

1. Automated static analysis
2. Dynamic analysis through emulation
3. Dynamic analysis through virtualization
4. Dynamic analysis through bare-metal (non VM)

Competition of NGAVs

ThreatTrack Security

NGAV Process



ThreatTrack Security

Unpackers In Next Gen AV

Pros

1. Augments existing detection.
2. Helps reduce processing files for dynamic analysis.
3. Provide faster scan time.

Cons

1. No use for APT or 0-day malware.
2. Applicable only to PE file format.

THE WINDOWS EXECUTABLE FORMAT

The EXE File

Assembly Code

```
ing 32-bit PE program as raw
format PE GUI
entry start

section '.text' code readable executable
 start:
    push    0
    push    _caption
    push    _message
    push    0
    call    [MessageBoxA]
    push    0
    call    [ExitProcess]

section '.data' data readable writeable
 _caption db 'Win32 assembly program',0
 _message db 'Hello World!',0
```

Compiled Binary

```
.00401000: 6A00          push    0
.00401002: 6800204000    push    000402000 ;'Win32 assembly program' --□1
.00401007: 6817204000    push    000402017 ;'Hello World!' --□2
.0040100C: 6A00          push    0
.0040100E: FF1544304000  call    MessageBoxA
.00401014: 6A00          push    0
.00401016: FF153C304000  call    ExitProcess
.0040101C: 0000          add     [eax],al
.0040101E: 0000          add     [eax],al
.00401020: 0000          add     [eax],al
.00401022: 0000          add     [eax],al
```

Equivalent C Pseudo code

```
MessageBoxA (0, "Hello World!", "Win32 assembly program", 0);
ExitProcess (0);
```

ThreatTrack Security

The EXE File

Memory Address | Hex Code | Disassembly

```
.00401000:  6A00          push    0
.00401002:  6800204000    push    000402000 ;'Win32 assembly program' --□1
.00401007:  6817204000    push    000402017 ;'Hello World!' --□2
.0040100C:  6A00          push    0
.0040100E:  FF1544304000  call    MessageBoxA
.00401014:  6A00          push    0
.00401016:  FF153C304000  call    ExitProcess
.0040101C:  0000          add     [eax],al
.0040101E:  0000          add     [eax],al
.00401020:  0000          add     [eax],al
.00401022:  0000          add     [eax],al
```

```
.00401000:  6A 00 68 00-20 40 00 68-17 20 40 00-6A 00 FF 15
.00401010:  44 30 40 00-6A 00 FF 15-3C 30 40 00-00 00 00 00
.00401020:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

CODE Section

```
.00402000:  57 69 6E 33-32 20 61 73-73 65 6D 62-6C 79 20 70   Win32 assembly p
.00402010:  72 6F 67 72-61 6D 00 48-65 6C 6C 6F-20 57 6F 72   rogram Hello Wor
.00402020:  6C 64 21 00-00 00 00 00-00 00 00 00-00 00 00 00   ld!
.00402030:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

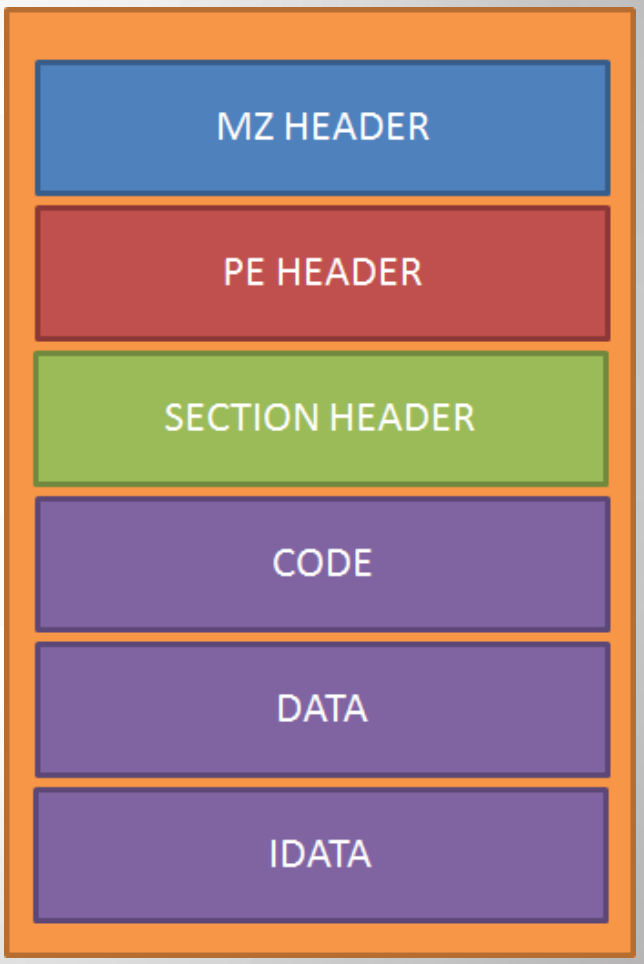
DATA Section

```
.00403000:  00 00 00 00-00 00 00 00-00 00 00 00-4C 30 00 00                L0
.00403010:  3C 30 00 00-00 00 00 00-00 00 00 00-00 00 00 00   <0
.00403020:  59 30 00 00-44 30 00 00-00 00 00 00-00 00 00 00   Y0  D0
.00403030:  00 00 00 00-00 00 00 00-00 00 00 00-64 30 00 00                d0
.00403040:  00 00 00 00-72 30 00 00-00 00 00 00-4B 45 52 4E       r0      KERN
.00403050:  45 4C 33 32-2E 44 4C 4C-00 55 53 45-52 33 32 2E   EL32.DLL USER32.
.00403060:  44 4C 4C 00-00 00 45 78-69 74 50 72-6F 63 65 73   DLL   ExitProces
.00403070:  73 00 00 00-4D 65 73 73-61 67 65 42-6F 78 41 00   s   MessageBoxA
```

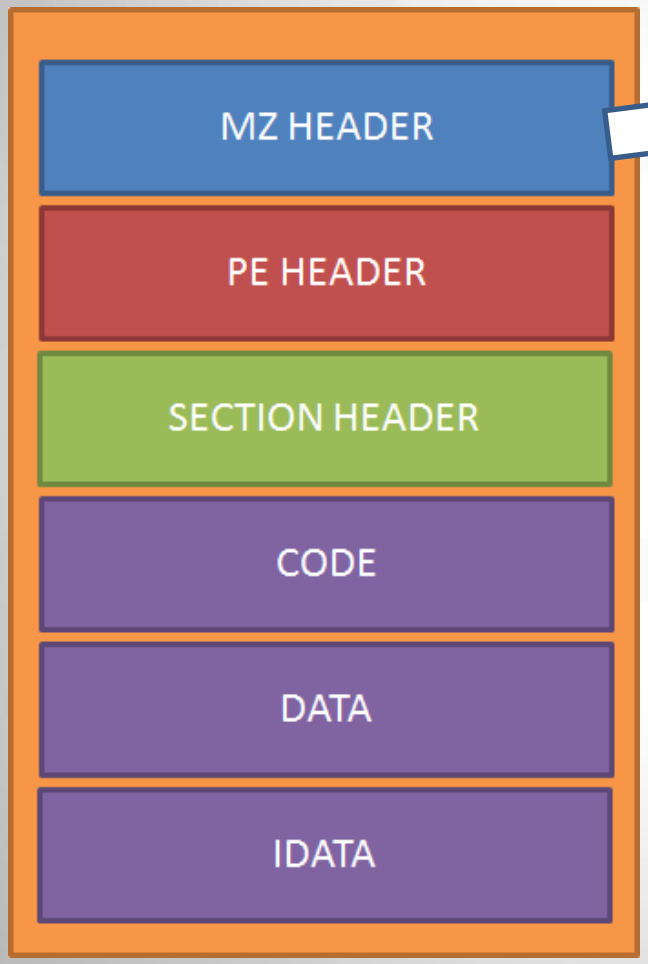
IDATA Section

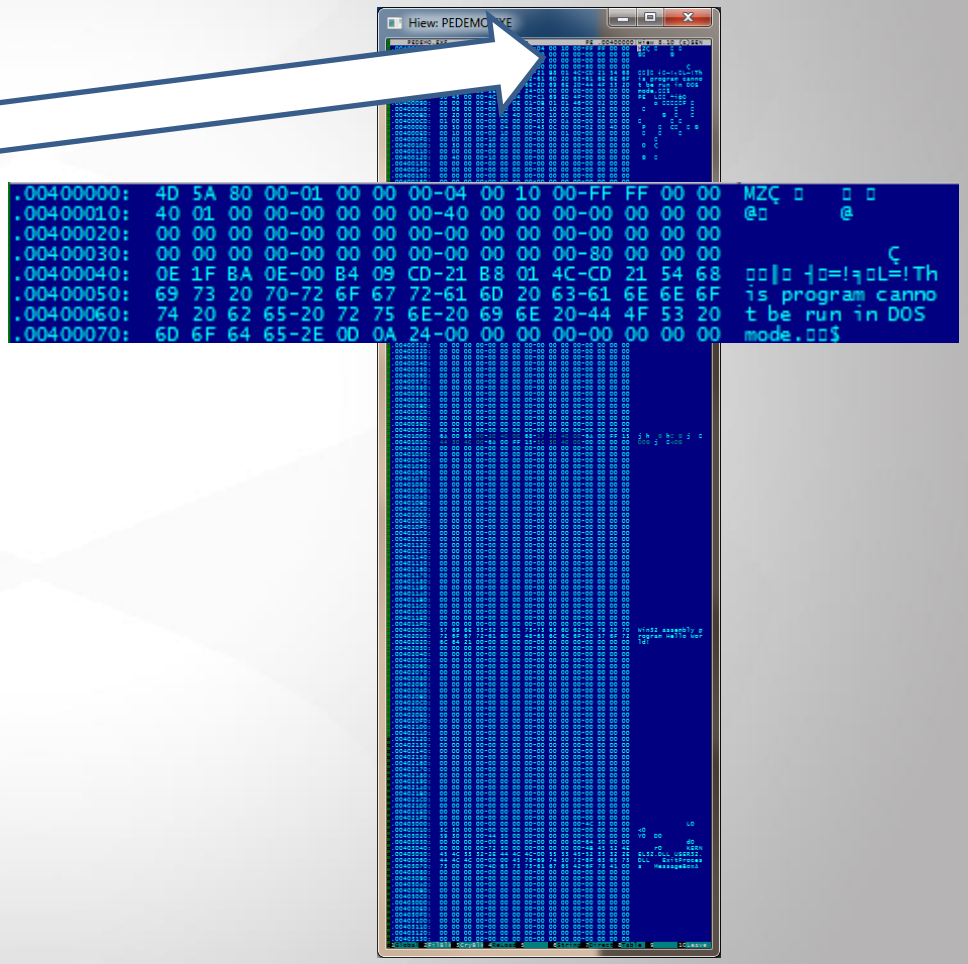
The EXE File

- EXE files are called Portable Executable files or PE file for short.
- PE format defines how a file should be structured to make it executable in Windows.
- SYS, DLL, OCX files are also PE files.



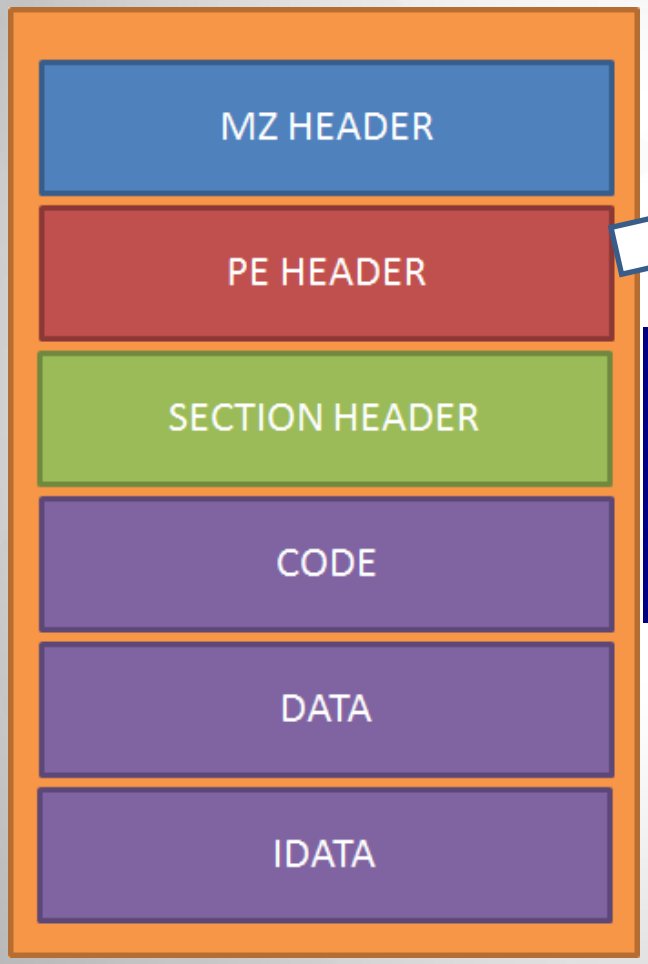
The EXE File

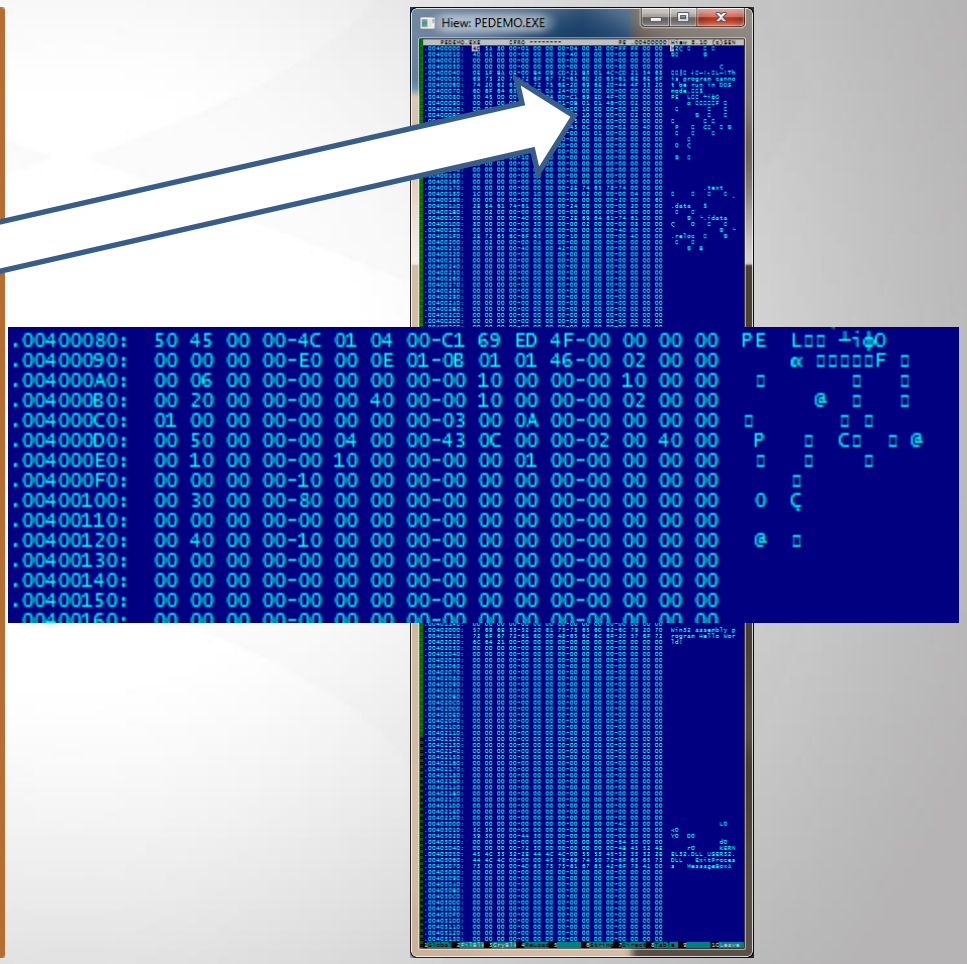




ThreatTrack Security

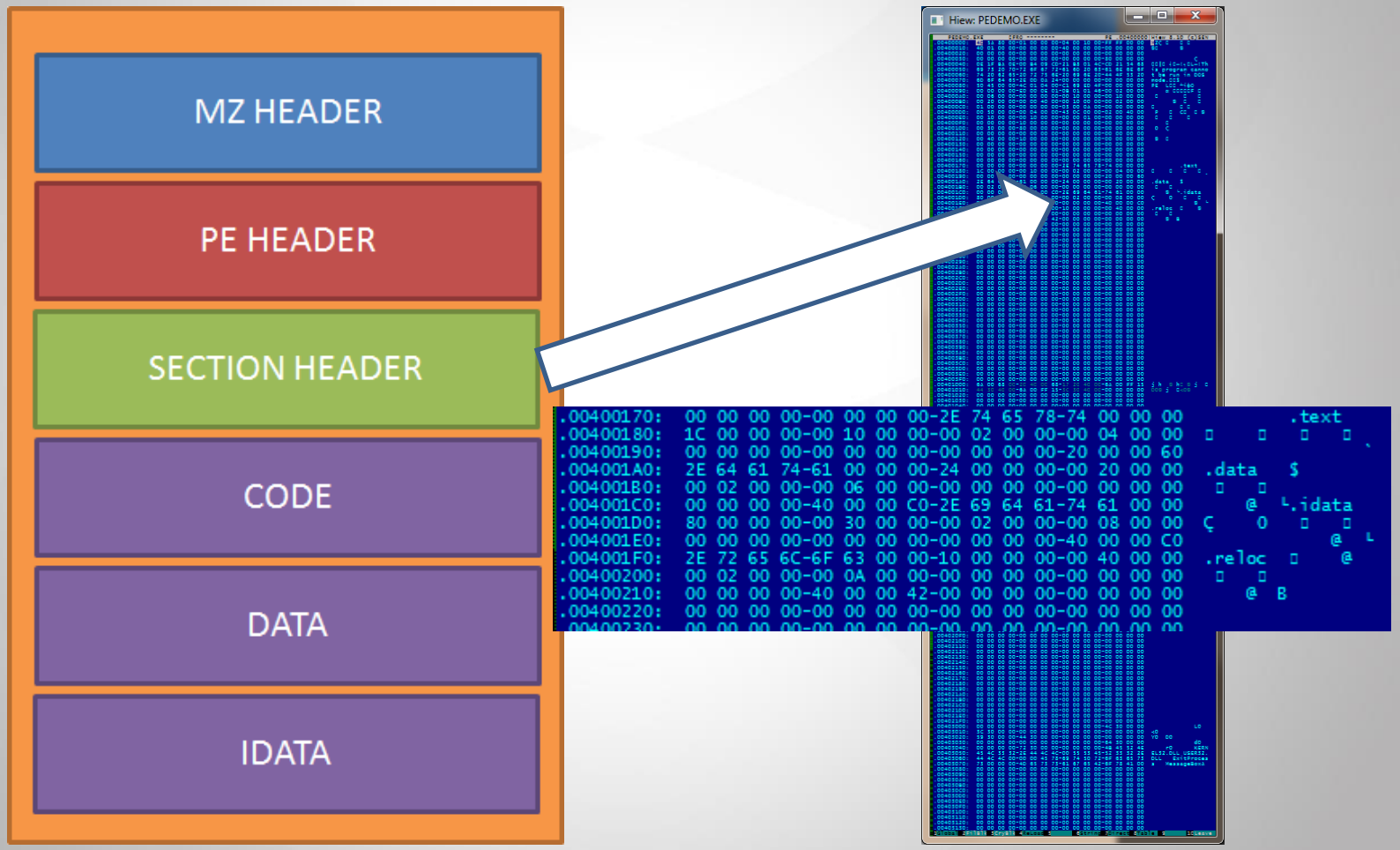
The EXE File





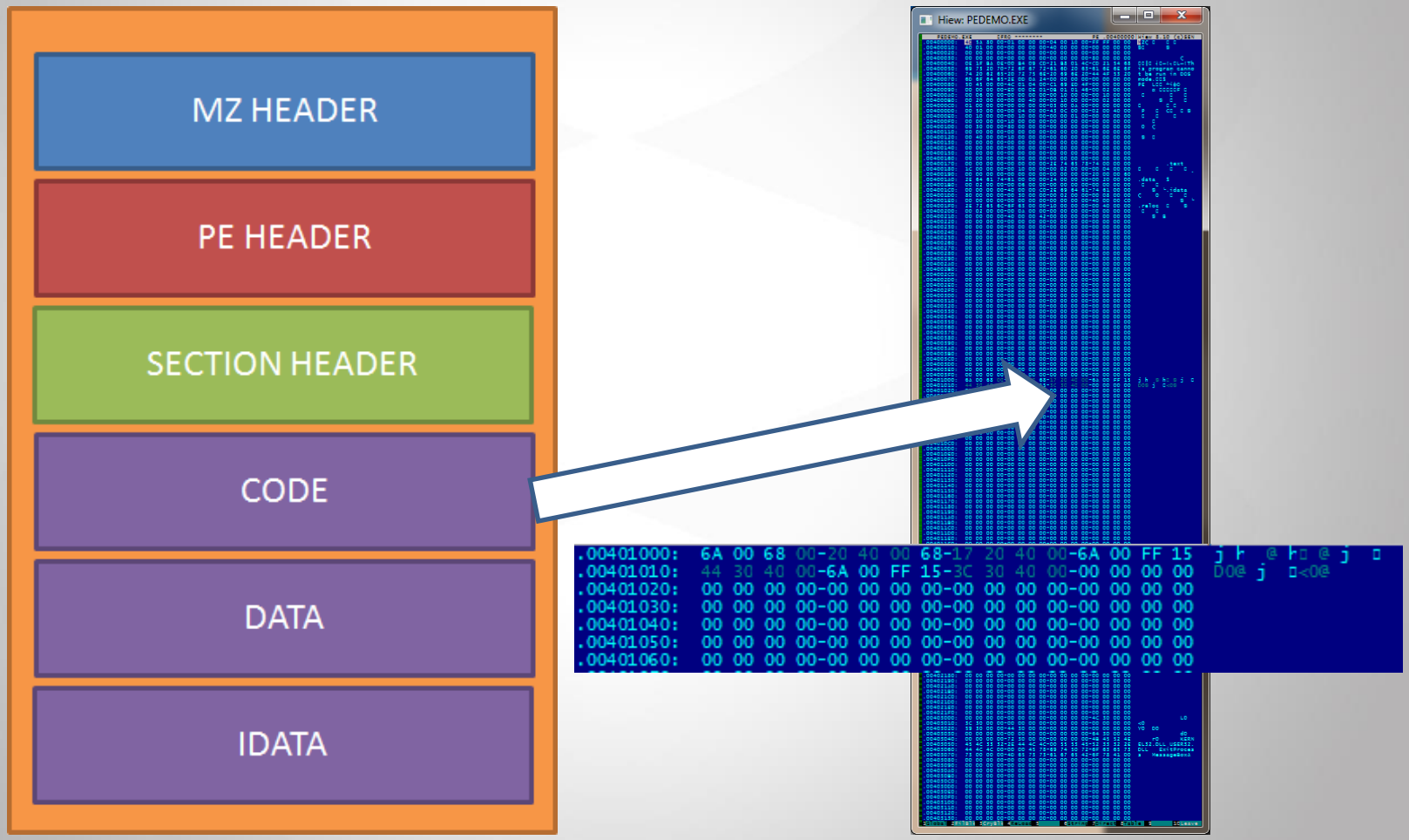
ThreatTrack Security

The EXE File

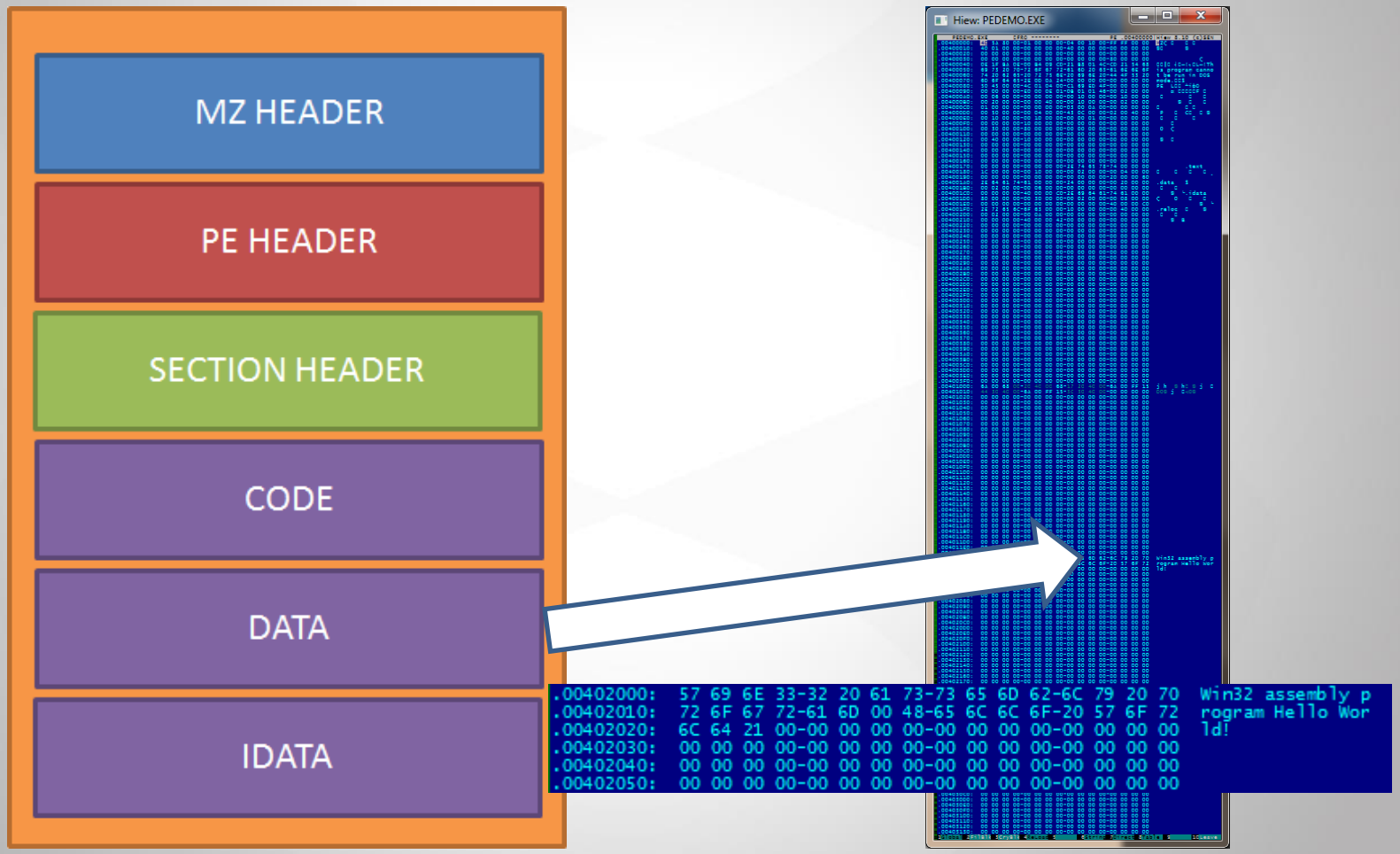


ThreatTrack Security

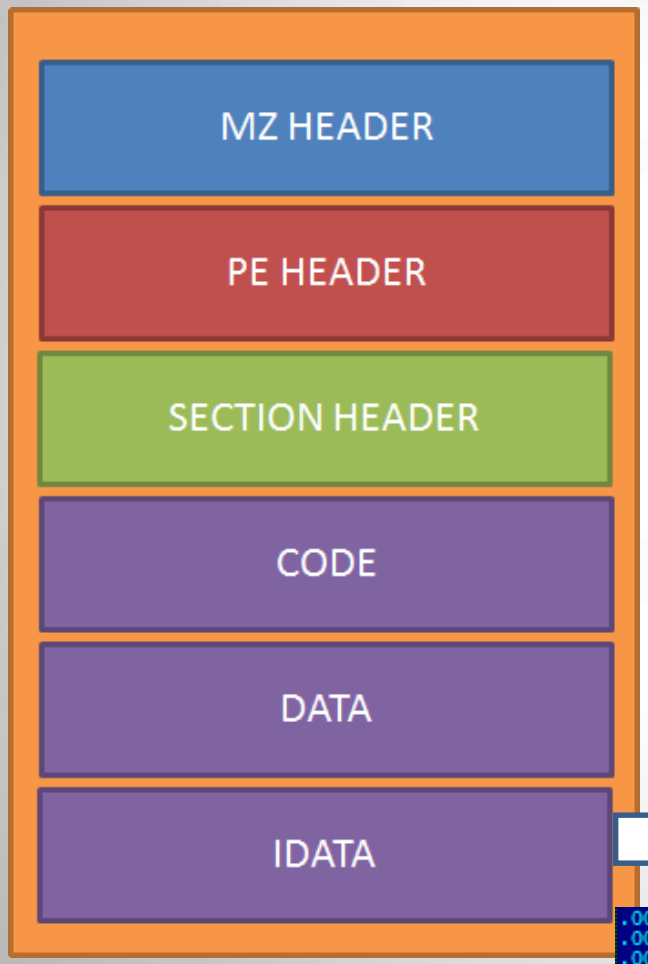
The EXE File

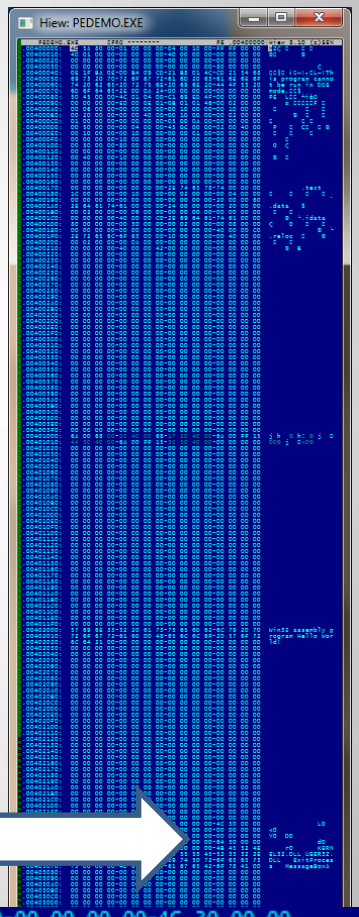

MZ HEADER
PE HEADER
SECTION HEADER
CODE
DATA
IDATA
Hiew: PEDEMO.EXE
.00401000: 6A 00 68 00-20 40 00 68-17 20 40 00-6A 00 FF 15
.00401010: 44 30 40 00-6A 00 FF 15-3C 30 40 00-00 00 00 00
.00401020: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00401030: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00401040: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00401050: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00401060: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00

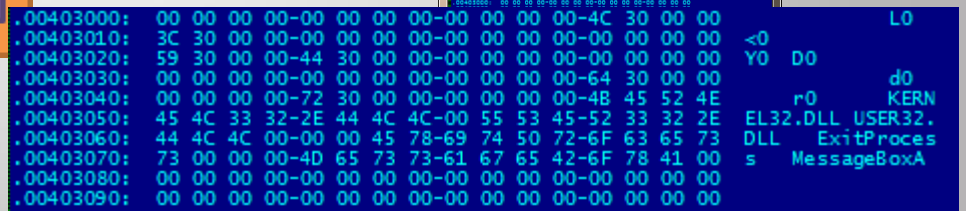
The EXE File



The EXE File

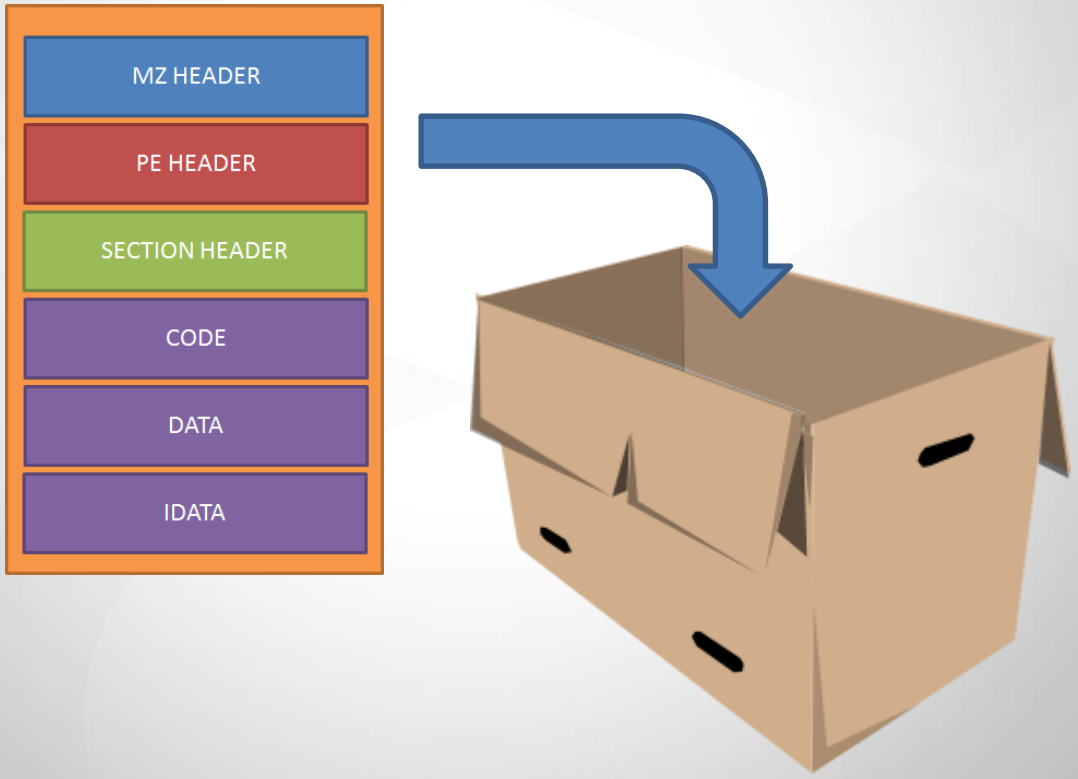






ThreatTrack Security

The EXE File



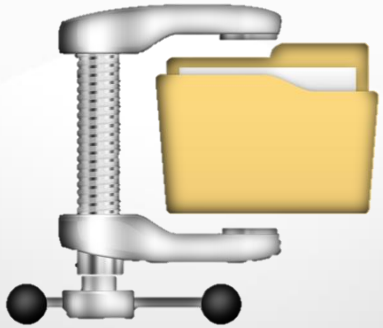
WHAT ARE PACKERS?

The PE Packer

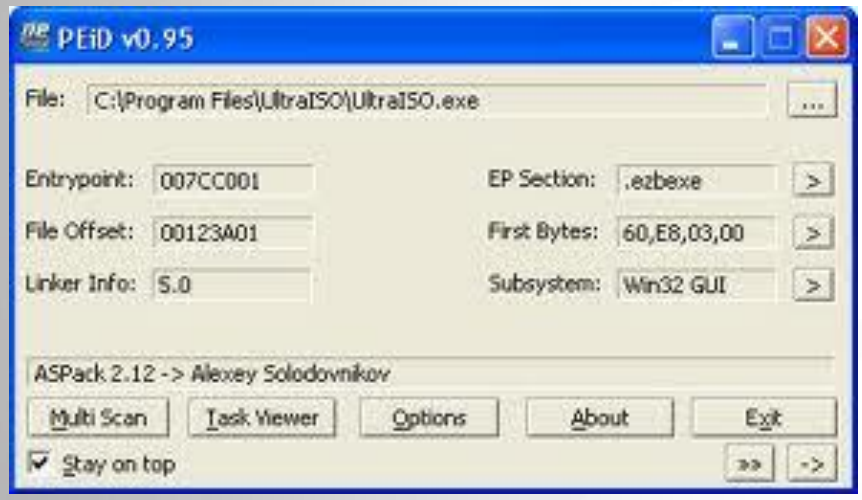
- Hides static codes from being inspected easily.
- Used by software companies to protect intellectual property.
- Used by malwares to avoid signature based detection.

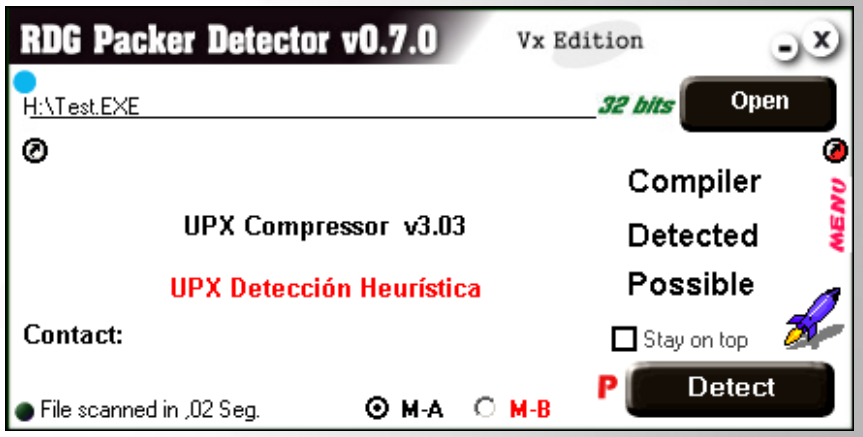
Types of Packer

1. Crypters – Uses keys to encrypt code and data.
2. Compressors – Uses compression algorithms to shrink code and data.
3. Protectors – Uses anti-debugging tricks to frustrate reversing.



Packer Detection





ThreatTrack Security

CREATING AN UNPACKER

When to Unpack?

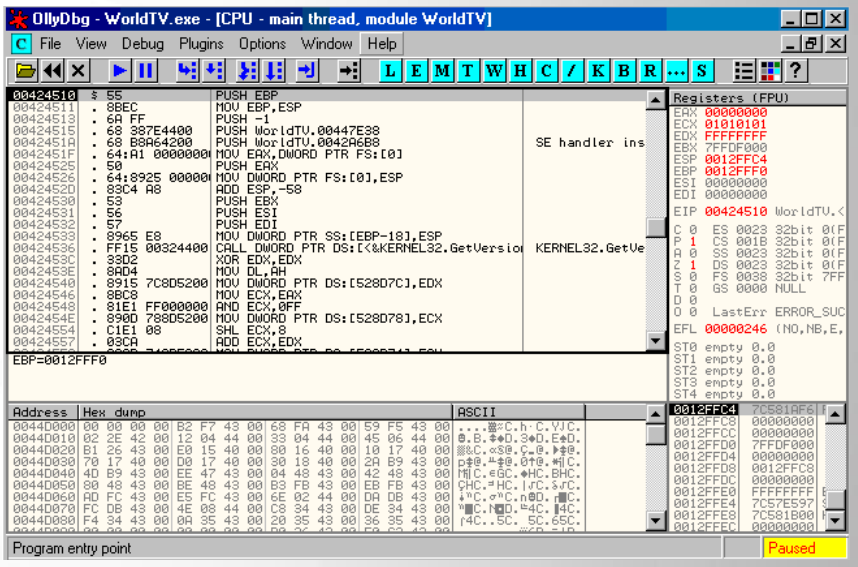
1. When a packer is used to defeat existing detection.
2. When a number samples from different malware families has a common packer stub.
3. When there’s a downloadable packer tool to create samples for testing.

What are Unpackers?

- Standalone applications, scripts, or add-on modules to a debugger or AV product.
- Restores code and data back to its original form.
- Expose code structure and strings for static analysis.

Unpacking Tools

- OllyDbg
  - OllyScript plugin
  - OllyDmp plugin
- Imprec
- LordPE



TitanMist

1. Open source
2. Based on OllyScript
3. LUA and Python supported
4. Created by ReversingLabs

TITANMIST DEMO

TitanMist Demo

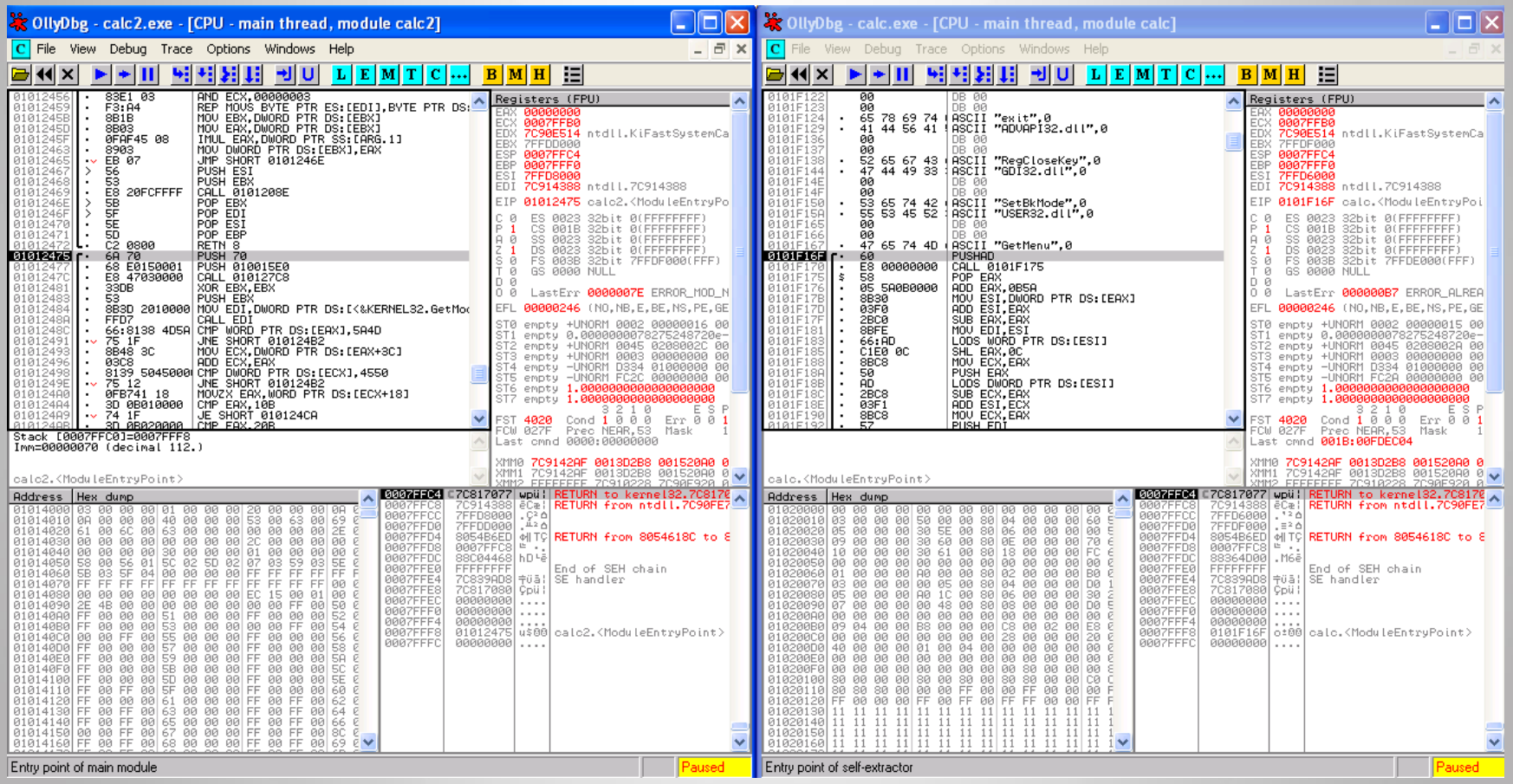
1. Python 2.7 (http://www.python.org/ftp/python/2.7/python-2.7.msi)
2. TitanMist 2.0 (http://www.reversinglabs.com/download/TitanMist.rar)
3. OllyDbg 2.0 (http://www.ollydbg.de/odbg200.zip)
4. MPRESS 2.17 packer (http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/MPRESS.shtml)
5. Host File: Win XP calc.exe (C:\Windows\System32\calc.exe)

TitanMist Demo

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\TitanMist>mpress.exe calc.exe

          MATCODE comPRESSor for executables
   Copyright (C) 2007-2010, MATCODE Software, MPRESS v2.17

<< calc.exe >>
PE32/x86 112.0kB -> 55.0kB  Ratio: 49.1%


C:\TitanMist>
```

TitanMist Demo



TitanMist Demo

db.xml - Notepad

File Edit Format View Help

```
                        89 85 ?? ?? ?? ?? 8B 85 ?? ?? ?? ?? 83 E0 01 74 48
                        8B FD 8D 85 ?? ?? ?? ?? 50 64 FF 35 ?? ?? ?? ?? 64
                        EB 01 FF CC 8B EF 64 8F 05 ?? ?? ?? ?? 83 C4 04 3C
                </signature>
                <signature start="ep" version="1.2">
                        60 E8 00 00 00 00 5D 81 ED ?? ?? ?? ?? B9 ?? ?? ??
                        ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
                        ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? AA E2 [7F-FF] -(1
                </signature>
                <signature start="ep" version="1.3">
                        55 8B EC 53 56 57 60 E8 00 00 00 00 5D 81 ED ?? ??
                        81 C2 ?? ?? ?? ?? 8D 3A 8B F7 33 C0 EB 04 90 EB 01
                        ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
                        ?? ?? ?? AA E2 [7F-FF] -(1) !(-2) AC
                </signature>
        </entry>
        <entry
                name="MPRESS"
                url="http://kbase.reversinglabs.com/index.php/Main_Page"
                description=""
                author="Matcode Software">
                <unpacker type="titanscript">
                        mpress.txt
                </unpacker>
                <signature start="ep" version="2.17">
                        60 E8 00 00 00 00 58 05 ?? ?? ?? ?? 8B 30 03 F0
                        2B C0 8B FE 66 AD C1 E0 0C 8B C8 50 AD 2B C8 03
                        F1 8B C8 57 51 49 8A 44 39 06 88 04 31 75 F6
                </signature>
        </entry>
</mistdb>
```

TitanMist Demo

```
C:\WINDOWS\system32\cmd.exe

C:\TitanMist\TitanMist>TitanMist.exe -i C:\TitanMist\calc.exe
        TitanMist by ReversingLabs Corp.
 version  :          1.0

 Match found!
 Name:         MPRESS
 Version:      2.17
 Author:       Matcode Software
 Wiki url:     http://kbase.reversinglabs.com/index.php/Main_Page
 Description:

eip: 01012475
 Unpacking succeeded!

C:\TitanMist\TitanMist>
```

FINAL THOUGHTS

- Unpackers are needed while packers exist.
- Unpacking is a valuable skill for malware analysts to learn.
- Antivirus products need unpackers to prolong the effectiveness of their signatures.
- Next Gen AV need unpackers/signature based detection to filter out non-APT or in-the-wild samples.

QUESTIONS?

References

1. Craig, P. (2006). Unpacking Malware, Trojans and Worms
2. Talekar, N. (2012). Practical Reversing II – Unpacking EXE
3. Albertini, A. (2010). Packers
4. Damballa (2011). Next Generation Anti-Virus